1981 - Issue 05

(C) Copyright 1981

POCKET COMPUTER NEWSLETTER

P.O. Box 232, Seymour, CT 06483

HEXED!

No, the TRS-80 PC doesn't have any real magical powers, at least none that I have discovered. "Hex" is short for *hexadecimal,* which is another way of denoting base 16. That is a numbering system near and dear to many a programmer's heart. Unfortunately, very few programmers are blessed with eight fingers per hand. Thus, unless they want to take off their shoes and socks, they have trouble counting in hexadecimal. Curses!

However, all is not lost. The pocket computer can change things to hexadecimal — once you show it how.

The accompanying routine converts decimal numbers to hexadecimal (label H) and hexadecimal numbers to decimal (label X). Since the PC can hardly handle groups of letters and since base 16 numbers use the letters A through F to represent the "digits" 10 through 15 respectively, you must key in hexadecimal numbers one digit at a time.

To convert 65535 to hexadecimal put the computer in the DEF mode (now it can't hear — sorry, I couldn't resist), type SHIFT-H and key 65535 ENTER. Following two beeps you see "65535 DEC" as verification of your input. Press ENTER and, a beep later, you will see "FFFF HEX".

To convert FFFF back to decimal, type SHIFT-X. The computer will ask for hexadecimal digits one at a time, going from left to right. Input "F ENTER" four times, then press the equal (=) key and ENTER for the result. "FFFF HEX" will be displayed after one beep to verify the input. Press the ENTER key one more time. Two beeps later you will see "65535 DEC" as the answer.

Please note the the somewhat unstructured arrangement of the program was purposely done to increase the speed of operation. The TRS-80 PC starts looking for labels and line numbers at the beginning of a program.

David Motto, 3639 Roosevelt, Jackson, MI 49203

```
1 "H" CLEAR:INPUT "DEC NUM?";M:M=INT ABS M:
  L=M:GOTO 36
2 "X" CLEAR:K=11:GOTO 38
3 A$(K)="0":RETURN
4 "0" L=16L:GOTO 38
5 A$(K)="1":RETURN
6 "1" L=16L+1:GOTO 38
7 A$(K)="2":RETURN
8 "2" L=16L+2:GOTO 38
9 A$(K)="3":RETURN
10 "3" L=16L+3:GOTO 38
11 A$(K)="4":RETURN
12 "4" L=16L+4:GOTO 38
13 A$(K)="5":RETURN
14 "5" L=16L+5:GOTO 38
15 A$(K)="6":RETURN
16 "6" L=16L+6:GOTO 38
17 A$(K)="7":RETURN
18 "7" L=16L+7:GOTO 38
19 A$(K)="8":RETURN
20 "8" L=16L+8:GOTO 38
21 A$(K)="9":RETURN
22 "9" L=16L+9:GOTO 38
23 A$(K)="A":RETURN
24 "A" L=16L+10:GOTO 38
25 A$(K)="B":RETURN
26 "B" L=16L+11:GOTO 38
27 A$(K)="C":RETURN
28 "C" L=16L+12:GOTO 38
29 A$(K)="D":RETURN
30 "D" L=16L+13:GOTO 38
31 A$(K)="E":RETURN
32 "E" L=16L+14:GOTO 38
33 A$(K)="F":RETURN
34 "F" L=16L+15:GOTO 38
35 "=" A$(K)="":GOSUB 39:GOSUB 40:GOTO "X"
36 K=K+1:N=M-16*INT(M/16):M=(M-N)/16:GOSUB 2N+3:
  IF M>0 THEN 36
37 GOSUB 40:GOSUB 39:GOTO "H"
38 K=K-1:PAUSE USING "###";11-K;"TH HEX CHAR?":
  INPUT A$(K):GOTO A$(K)
39 BEEP 2:PRINT USING "##########";L;" DEC":
  RETURN
40 BEEP 1:PRINT J$;I$;H$;G$;F$;E$;D$;C$;B$;A$;" HEX":
  RETURN
```

MORE OPERATING TIPS FOR THE TRS-80 PC

FOR loops are relatively slow on the Radio Shack TRS-80 PC. This is especially noticeable when using nested loops. Also, watch out for the 4 loop maximum!

The statement label capability of the TRS-80 PC is great, but GOSUBing to labels is much, much slower than using line numbers. In a particular application I was programming involving matrix manipulations, I saved 1/3 the program execu-

tion time by converting from labels to line numbers. If the application is short with few iterations of code, the difference may not amount to much. However, it can make a great deal of difference in a highly modularized program.

If you have several lines of similar code in a program, such as:

```
100 IF A=1 THEN 500
110 IF A=2 THEN 600
120 IF A=3 THEN 700
```

You can save a lot of keying by editing the line number and other differences after you have entered the original line. Since editing is done in a buffer, the original line is undisturbed. Just make the changes, hit ENTER, and presto!

Using the procedure discussed in the preceeding paragraph can give you the capability of a quasi-renumbering operation. Simply LIST the line, edit the line number and press ENTER. A few notes of caution are in order here. First, remember that you still have to manually change all line number references such as GOSUBs, etc. Second, don't forget that the original line still exists. To erase it, if desired, just key in the line number and hit ENTER.

When you desire to delete a series of lines in a program, start with the highest line number. This will save time. Why? Because making changes in the middle of program memory (whether editing or deleting) necessitates shifting the contents of memory (to the end of program storage) to accomodate the changes. Working at the end of the program reduces the amount of such manipulation.

The TRS-80 PC manual does not explicitly state that the PRINT USING statement does not perform rounding on the pocket computer as it does in some other dialects. On the TRS-80 PC it simply truncates or adds zeros. Remember to use a round-off algorithm, if necessary, when using the PRINT USING statement!

When editing numeric expressions or anything not enclosed by quotation marks, it is quicker to use the SPC key than the SHIFT DEL keys. When you hit the ENTER key the interpreter will remove any unnecessary spaces!

Submitted by: *C. C. Vans Evers II, 3607 Oklahoma, Tampa, FL 33611*

THE TAX MAN COMETH – AGAIN!

Issue 03 of *PCN* carried a neat program that calculated payroll deductions and net pay for employees paid on a weekly basis. One of our readers has written an even "neater" version of the program that we thought you might appreciate – so here it is. Thanks to: *David Motto, 3639 Roosevelt, Jackson, MI 49203*

```
10 USING "#####.##":H=E2
20 G=0:INPUT "ENTER GROSS PAY? ";G
30 E=0:INPUT "# OF EXEMPTIONS? ";E
40 M=0:INPUT "SINGLE(0)/MARRIED(1)? ";M:M=INT
   ABS M
50 IF (M<0)+(M>1) GOTO 40
60 X=0:T=G-19.23E:GOSUB 600+200M
70 F=.0665G
80 BEEP 2:X=INT (HX+.5)/H:PRINT "FWT= ";X
90 F=INT (HF+.5)/H:PRINT "FICA= ";F
100 Z=0:INPUT "OTHER WITHOLDINGS? ";Z
110 Z=INT (HZ+.5)/H
120 G=G-X-F-Z:PRINT "NET PAY= ";G
130 GOTO 20
600 IF T>=433 LET X=103.39+.39*(T-433):RETURN
610 IF T>=331 LET X=68.71+.34*(T-331):RETURN
620 IF T>=273 LET X=51.31+.3*(T-273):RETURN
630 IF T>=196 LET X=31.29+.26*(T-196):RETURN
640 IF T>=131 LET X=17.64+.21*(T-131):RETURN
650 IF T>=63 LET X=5.4+.18*(T-63):RETURN
660 IF T>=27 LET X=.15*(T-27)
670 RETURN
800 IF T>=556 LET X=119.35+.37*(T-556):RETURN
810 IF T>=454 LET X=86.71+.32*(T-454):RETURN
820 IF T>=369 LET X=62.91+.28*(T-369):RETURN
830 IF T>=288 LET X=43.47+.24*(T-288):RETURN
840 IF T>=210 LET X=27.09+.21*(T-210):RETURN
850 IF T>=127 LET X=12.15+.18*(T-127):RETURN
860 IF T>=46 LET X=.15*(T-46)
870 RETURN
```

JACKPOT SLOTS

Here is a very nicely written game for the Radio Shack TRS-80 Pocket Computer. Watch out – you can find yourself spending a lot of time with this game!

Jackpot Slots was submitted by: *Gary Heidbrink, Box 24, FPO New York, NY 09593*

```
5 "J":Z=3254:H=0:O=0:J=0
10 A$=" CHERRY"
15 B$=" APPLE"
20 C$=" PLUM"
25 D$=" BELL"
30 E$=" BAR"
35 F$=" 777"
40 G$=" ANY"
45 PRINT "    JACKPOT SLOTS":W=0
50 INPUT "NEED INSTRUCTIONS?(Y/N)";M$.IF M$=
   "N" GOTO 105
55 IF M$="J" PRINT "TAKE:$";J;" BAL:$";O:M$=
   "Y":GOTO 45
60 PRINT " EACH PLAY COSTS $ .25"
65 PRINT " PAY OFFS ARE:"
70 PRINT A$;G$;G$;"  $ .50"
75 PRINT A$;A$;G$;" $1.25"
80 PRINT B$;B$;B$;" $2.50"
85 PRINT C$;C$;C$;"  $3.50"
90 PRINT D$;D$;D$;"  $6.75"
95 PRINT E$;E$;E$;"  $50.00"
100 PRINT F$;F$;F$;" MIN. $50.00":PAUSE ""
105 INPUT " CHOOSE A MACHINE:";Z
110 INPUT " PLAY OR QUIT? (P/Q):";M$:IF M$="Q"
    GOTO 290
115 PAUSE " LEVER HAS BEEN PULLED!":J=J+.25:
    H=H+.05:W=W-.25:GOTO 125
```

```
120 Z=439147*Z+J:Z=23*Z-INT(Z*.23)*100:Y=INT
    (Z*.2)+1:RETURN
125 GOSUB 120
130 IF Y<8 LET I$=C$:GOTO 160
135 IF Y<14 LET I$=B$:GOTO 160
140 IF Y<17 LET I$=E$:GOTO 160
145 IF Y<19 LET I$=A$:GOTO 160
150 IF Y=19 LET I$=D$:GOTO 160
155 I$=F$
160 GOSUB 120
165 IF Y<7 LET K$=A$:GOTO 195
170 IF Y<12 LET K$=D$:GOTO 195
175 IF Y<16 LET K$=B$:GOTO 195
180 IF Y<19 LET K$=C$:GOTO 195
185 IF Y=19 LET K$=F$:GOTO 195
190 K$=E$
195 GOSUB 120
200 IF Y<10 LET L$=D$:GOTO 225
205 IF Y<15 LET L$=B$:GOTO 225
210 IF Y<19 LET L$=C$:GOTO 225
215 IF Y=19 LET L$=E$:GOTO 225
220 L$=F$
225 BEEP 1:PAUSE I$;" ";K$;" ";L$
230 IF I$=K$ GOTO 245
235 IF I$=A$ LET X=.5:GOTO 285
240 PAUSE " YOU LOSE!!":GOTO 110
245 IF I$=A$ LET X=1.25:GOTO 285
250 IF I$=L$ GOTO 260
255 GOTO 240
260 IF I$=B$ LET X=2.5
265 IF I$=C$ LET X=3.5
270 IF I$=D$ LET X=6.75
275 IF I$=E$ LET X=50
280 IF I$=F$ LET N=INT(H/.25)*.25:X=N+50:H=H-N
285 PRINT " YOU WIN!!  $";X:W=W+X:GOTO 110
290 IF W<0 GOTO 300
295 PRINT " HOUSE OWES YOU $";W:O=O-W:GOTO 45
300 W=W*-1:PRINT " YOU OWE $";W:O=O+W:GOTO 45
```

TRS-80 POCKET COMPUTER TOKENS

Each byte of memory in the Radio Shack TRS-80 PC that is used for program storage of a user program contains a two-digit hexadecimal number. This number represents a particular character or instruction. The words making up the vocabulary of the computer, such as STEP, GOTO, INT, RETURN, and so forth, are thus *tokenized.* That is, each word is stored as a single byte that the interpreter is able to recognize as a symbol for the instruction or command. An instruction such as INPUT, though it may have been typed in by the user as a six letter word, is simply stored as a single byte after the program line is accepted. This process of *tokenizing* allows programs to be greatly compacted when stored in memory.

Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158, has spent a great deal of time and effort determining the tokens used in the TRS-80 PC. The hexadecimal tokens he has figured out are shown in the accompanying table.

It may be noted that the hexadecimal representation of the numeral digits 0 to 9 and the letters of the alphabet correspond with standard ASCII codes. However, punctuation marks and other symbols do not.

It is also interesting to note that a program line number begins with E0 to E9 as its first byte and 00 to 99 as its second byte. (Thus representing line numbers from 000 through 999.) A line in memory ends with the null byte, 00, which is produced by the ENTER key.

00	ENTER	5A	J	B4	CONT
11	SPC	5B	K	B5	DEBUG
12	"	5C	L	B6	CSAVE
13	?	5D	M	B7	CLOAD
14	!	5E	N		
15	#	5F	O	C0	GRAD
16	%	60	P	C1	PRINT
17	¥	61	Q	C2	INPUT
18	$	62	R	C3	RADIAN
19	π	63	S	C4	DEGREE
1A	√	64	T	C5	CLEAR
1B	,	65	U		
1C	;	66	V	D0	IF
1D	:	66	V	D1	FOR
		67	W	D2	LET
30	(	68	X	D3	REM
31	)	69	Y	D4	END
32	>	6A	Z	D5	NEXT
33	<			D6	STOP
34	=	82	>=	D7	GOTO
35	+	83	<=	D8	GOSUB
36	-	84	<>	D9	CHAIN
38	/			DA	PAUSE
39	Δ	90	TO	DB	BEEP
40	0	91	STEP	DC	AREAD
41	1	92	THEN	DD	USING
42	2			DE	RETURN
43	3	A0	SIN		
44	4	A1	COS	E0	Begin Line 0--
45	5	A2	TAN	E1	Begin Line 1--
46	6	A3	ASN	E2	Begin Line 2--
47	7	A4	ACS	E3	Begin Line 3--
48	8	A5	ATN	E4	Begin Line 4--
49	9	A6	EXP	E5	Begin Line 5--
4A	.	A7	LN	E6	Begin Line 6--
4B	E	A8	LOG	E7	Begin Line 7--
		A9	INT	E8	Begin Line 8--
51	A	AA	ABS	E9	Begin Line 9--
52	B	AB	SGN		
53	C	AC	DEG		
54	D	AD	DMS		
55	E				
56	F	B0	RUN		
57	G	B1	NEW		
58	H	B2	MEM		
59	I	B3	LIST		

PROGRAM: _____ DATE: _____ PAGE___ of___

STEP		VARIABLE USAGE		
		A	1	
		B	2	
		C	3	
		D	4	
		E	5	
		F	6	
		G	7	
		H	8	
		I	9	
		J	10	
		K	11	
		L	12	
		M	13	
		N	14	
		O	15	
		P	16	
		Q	17	
		R	18	
		S	19	
		T	20	
		U	21	
		V	22	
		W	23	
		X	24	
		Y	25	
		Z	26	

NOTES:

TIC-TAC-TOE FOR THE RADIO SHACK TRS-80 PC

Here is a listing of a program that plays a non-losing game of Tic-Tac-Toe and keeps track of the score.

```
1 E=34:F=24:G=72:I=23:J=42:K=43:L=K:M=E:O=16:P=13
  :R=31:S=61:T=17:U=K:V=0:W=0
2 H=0:Q=0:GOSUB 11:IF X=5 GOTO 20
3 A=X:B=X-2*INT(X/2:Y=5:GOSUB 10:C=X:GOSUB 7
  :D=Y:N=A(D+12-9B):C=INT.1N:GOSUB 7:GOSUB 10
  :IF X+Y<>10 GOTO 13
4 C=N-10*INT.1N:GOSUB 7:GOSUB 10:IF X+Y<>10
  GOTO 13
5 IF B=0 IF D=8 LET C=6:GOSUB 7:GOTO 13
6 C=8+B*(D<9:GOSUB 7:GOTO 14
7 Y=INT((C+2)/3:Z=5-ABS(5-A:IF Z=3 LET C=6Y-C-2
8 IF Z=4 LET C=3C-8Y+6
9 Y=ABS(C-10*(A>5:RETURN
10 USING "##";PAUSE "MY MOVE,";Y:Q=10Q+Y
11 X=0:INPUT "YOUR MOVE: ";X:H=10H+X:RETURN
12 GOSUB 30:GOTO 11
13 W=W+V+1:V=0:Y=10-Y:Q=10Q+Y:BEEP 1:PAUSE
   "I WIN IN SQUARE";Y:GOTO 15
14 V=V+1:Q=10Q+Y:PAUSE "MY MOVE,";Y;"; TO
   KITTY."
15 PAUSE USING "###";"THE SCORE:":PRINT "YOU 0;
   ME";W;"; KITTY";V:GOTO 2
20 Y=1:GOSUB 10:A=X:Y=10-X+6*(X=9:GOSUB 10:B=X
   :IF ABS(A-3)=1 GOTO 25
21 Y=9-A+6*(A>6:IF X+Y<>10 GOTO 13
22 GOSUB 10:Y=A-4:IF B/2>INT(B/2 IF X+Y<>10
   GOTO 13
23 Y=6-X:IF X/2=INT(X/2 LET Y=10-X
24 GOTO 14
25 Y=10-B+(5-A)*(B=9:GOSUB 10:IF ABS(A-B+2)=1
   GOTO 27
26 Y=10-X+(X=9)*(X-8+ABS(B-3:GOTO 14
27 Y=1+(A+3)*(X=9:GOTO 13
30 PRINT USING "#####";"YOU:";H;"    ME:";Q:
   USING "##":RETURN
31 "D" GOSUB 30:GOTO 2
```

The spaces of the playing board are designated by number. The numbers correspond to their positions on the numeric keyboard of the pocket computer:

7	8	9
4	5	6
1	2	3

Execution of the program is begun by RUN, ENTER. Then, enter each of your moves when prompted by the display, entering the number of the space in which you want your mark. The computer will respond with its move. After each game, the running score is displayed. Key ENTER to begin the next game.

If you forget what marks have been made during the course of a game, key ENTER when it is your turn *without* having input your mark. The computer will display the marks made thus far by both you and it. Key ENTER again to resume play. (At the end of a game, key SHIFT D to display the marks made in the game just completed; keying ENTER will start the next game.)

This program was submitted by: *Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158.*

A NUMERIC SORT

The accompanying Shell/Walters sort routine is a reasonably fast sort, even on the Radio Shack TRS-80 Pocket Computer. A 25-item random list was sorted in 110 seconds. That sorted list was resorted in reverse order in 101 seconds. This might be accomplished faster by hand. However, as a subroutine in a program, this amount of time might not be out of line considering the other types of operations that can be done by the PC at a much faster rate than manual computations!

```
1 REM NUMERIC SORT
10 CLEAR:PAUSE "NUMERIC SORT":INPUT "HOW
   MANY NUMBERS?";N:K=26
11 FOR I=1 TO N:PAUSE "#";I:INPUT "> ";B:A(I+K)=B:
   NEXT I
12 PAUSE "SORT BEGINNING..."

988 REM SORT ROUTINE
989 BEEP 1:C=N
990 C=INT(C/3)+1:FOR I=1 TO N-C:IF A(I+K)<=A(I+C
    +K) THEN 996
991 T=A(I+C+K):B=I
992 A(B+C+K)=A(B+K):B=B-C
993 IF B>0 IF T<A(B+K) THEN 992
995 A(B+C+K)=T
996 NEXT I:IF C>1 THEN 990
998 BEEP 3:FOR I=1 TO N:S=A(I+K):PRINT "N(";I;")= "
    ;S:NEXT I
999 PAUSE "END OF RUN...":END
```

A(I) holds the array that is to be sorted starting at A(27). This puts the array up out of the way of other variables used by the sort itself as well as those used by any (user supplied) main program. N is the size of the array. K sets the beginning point of the array. N and K must be passed to the sort routine. K could be part of the sort routine if there were no reason to change it. I have found it easier to initialize K in the main program.

The maximum size (N) of the array is dictated by the amount of other resident code. Typing MEM will display how many flexible memories are available.

Note that this is strictly a *numeric* sort. The only string handling command for the TRS-80 PC is an equality (=) test. Less-than's and greater-than's are not supported. (If you want to use an IBM 370, buy one!)

This sort can be useful in any applications in which a list of numbers must be sorted (dates, invoice numbers, etc.). With a little programming effort, one could set up indexing

for strings (7 characters only in the TRS-80 PC), but data memory capacity is so limited as to make such efforts essentially academic exercises.

Incidentally, this sort is faster than a Heap-sort that I converted. I think the primary reason is that the Heap-sort used two separate FOR loops rather than one. FOR loops chew up run time on the TRS-80 PC.

C. C. Vans Evers II, 3607 Oklahoma, Tampa, FL 33611

FINDING PRIME FACTORS OF A POSITIVE INTEGER

The program shown here for a Radio Shack TRS-80 PC lists the prime factors of any positive integer less than 1E10.

```
10 INPUT "INTEGER: ";I:J=I:M=0:T=2:GOSUB 50:T=3:
   GOSUB 50:T=5:GOSUB 50:T=1
20 T=T+6:GOSUB 50:T=T+4:GOSUB 50:T=T+2:GOSUB 50:
   T=T+4:GOSUB 50:T=T+2:GOSUB 50
30 T=T+4:GOSUB 50:T=T+6:GOSUB 50:T=T+2:GOSUB 50:
   IF TT<=J GOTO 20
40 GOTO 100
50 IF J/T-INT(J/T RETURN
60 IF T=I GOTO 130
70 M=M+1:J=J/T:IF J/T=INT(J/T GOTO 70
80 BEEP 1:PRINT USING "##########";T;" IS A
   FACTOR":PRINT USING "###";" OF MULTIPLICITY"
   ;M
90 M=0:IF TT<=J RETURN
100 IF I=J GOTO 130
110 IF J=1 PRINT " END OF LIST":END
120 T=J:J=1:M=1:GOTO 80
130 BEEP 1:PRINT USING "##########";I;" IS A
    PRIME":END
```

An Example

To factor 13510750, which equals $2x5^3x11x17^3$, key in RUN. When "INTEGER: " is displayed, input 13510750 and the ENTER key, which begins (and is used to continue) program operation. The results are shown thus:

```
 2 IS A FACTOR    OF MULTIPLICITY 1
 5 IS A FACTOR    OF MULTIPLICITY 3
11 IS A FACTOR    OF MULTIPLICITY 1
17 IS A FACTOR    OF MULTIPLICITY 3
  END OF LIST
```

Program Notes

The program is designed to hold down execution time. Multiples of 2, 3 and 5 (other than 2, 3 and 5 themselves) are omitted as trial divisors. The worst case, with 9999999967 as input, takes five hours; the number is a prime.

The program could be slightly shortened and speeded up by restricting usable inputs to smaller numbers, permitting tests of divisibility *not* requiring the use of the internally calculated 11th and 12th "guard" digits.

Variables Assignments

I = Integer to be factored
J = Given integer reduced by factors found to be prime
T = Test factor
M = Multiplicity count

This program was submitted by: *Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158.*

THANKS, ROBERT!

The programming form included on page 4 of this issue is a slightly modified version of one submitted by *Robert J. Leahy of 15742 Williams Street, Tustin, CA 92680.* Note that there is no other material on that page so that you may reproduce the form in quantity for your own use!

FROM THE WATCH POCKET

No news yet on the release of the hand held computers being planned by Panasonic/Quasar. Some trade journal reports are pegging a "system" based on these units in the $2,000.00 price range.

Reports are now circulating that Sharp will introduce its PC-1211 Pocket Computer — virtually identical to the TRS-80 PC (after all, it is widely assumed that they built the R. S. unit) — to the United States market in April or early May. Europeans are already familiar with the unit. Sharp also has the CE-122 combination printer and cassette interface ready to sell with the unit. The printer offers 16 characters per line. The unit is battery operated or can be used with an optional a.c. adapter.

The Sharp unit is reported to come with three manuals. One of these is a 300+ page "Applications Manual" which contains well over 100 programs. These range over the subject areas of mathematics, statistics, geometry, civil and electrical/electronic engineering, business and finance as well as a number of popular games. This manual would seem very attractive to current TRS-80 PC owners — keep an eye open for it.

There are now several publishers producing programs and booklets for PC users. One outfit, *Sebree's Computing, 456 Granite Avenue, Monrovia, CA 91016,* has a version of "Wumpus Adventure" available on cassette for $7.95 plus 95 cents shipping/handling. Another firm, *ARCsoft Publishers, PO Box 132, Woodsboro, MD 21798,* is working on a line of programs in booklet form and has pads of programming forms. Write to them for a catalog if interested.

SHARE YOUR POCKET COMPUTER KNOWLEDGE

We are always interested in hearing about applications for pocket computers, new products and news of the industry. Do you know of any operating time-savers, programming tips, short cuts or technical information that might be of interest to others?

If so, write it up! If your submission is accepted for publication you will earn a cash reward. If we cannot use your material, we will endeavor to promptly return it *provided you include a self addressed stamped envelope for that purpose.* We are careful, but we cannot assume responsibility for submitted materials.

Submissions should be addressed to: The Editor, PC NEWSLETTER, PO Box 232, Seymour, CT 06483